TCG Storage Workgroup Security Subsystem Class: Optical

Specification Version 1.0

2008 September 25

Contacts:

optical_storage@trustedcomputinggroup.org

TCG

Acknowledgements

The Optical Storage Sub-working Group of the TCG Storage Working Group wishes to thank all those who contributed to this specification.

A special thank you goes to Robert Thibadeau, TCG Storage Working Group chair and the members of the TCG Storage Working Group for their invaluable contributions and assistance.

William McFerrin,

Editor and Optical Storage Sub-working Group chair

Contents

Tables

Figures

1 Introduction

1.1 Document Purpose

TCG Storage Workgroup specifications provide a comprehensive architecture for putting storage devices under policy control as determined by the trusted platform host, by the capabilities of the storage device to conform with the policies of the trusted platform, and by the lifecycle state of the storage device as a Trusted Peripheral (TPer).

The Storage Architecture Core Specification [SA Core] is the guiding document for TCG Storage Workgroup specifications. The TCG Storage Workgroup publishes Security Subsystem Class (SSC) specifications that define storage device classes that select only those [SA Core] features that pertain to the use cases and threat models of a class.

1.2 Scope and Audience

This Optical Security Subsystem Class [OSSC] specification limits [SA Core] features to only those that pertain to optical storage use cases and threat models. Primarily, this [OSSC] addresses the data-at-rest problem of lost or stolen optical discs. It defines a [SA Core] minimally compliant optical TPer. Part of the optical TPer is formed within the device and another part is transported on an optical disc. It includes features that allow optical storage to support the security policies of individuals and organizations.

The intended audiences for this document is optical drive manufacturers, software developers, and authentication services.

2 References

- [SPC-3] ANSI NCITS 408-2005, SCSI Primary Commands – 3
- [SPC-4] T10/1731D, SCSI Primary Commands – 4
- [MMC-5] ANSI NCITS 430-2007, Multi-Media Commands – 5 (MMC-5)
- [MMC-6] T10/1836D Multi-Media Commands – 6 (MMC-6) Draft
- [FIPS140-2] National Institute of Standards and Technology (NIST), Security Requirements for Cryptographic Modules, FIPS Pub 140-2, 2001 May
- [FIPS140-3] National Institute of Standards and Technology (NIST), Security Requirements for Cryptographic Modules, FIPS Pub 140-3 Draft, 2007
- [FIPS180-3] National Institute of Standards and Technology (NIST), Secure Hash Standard (SHS) FIPS Pub 180-3, 2007 June
- [FIPS 197] National Institute of Standards and Technology (NIST), Standard for Advanced Encryption Standard (AES), FIPS Pub 197, 2001 November
- [FIPS198-1] National Institute of Standards and Technology (NIST), The Keyed-Hash Message Authentication Code (HMAC), FIPS Pub 198-1, 2007 June
- [NIST SP800-38A] National Institute of Standards and Technology (NIST), Recommendation for Block Cipher Modes of Operation - Methods and Techniques, NIST Special Publication 800-38A, 2001 December
- [NIST SP800-56A] National Institute of Standards and Technology (NIST), Recommendation for Pair-Wise Key Establishment Schemes Using Discreet Logarithm Cryptography, NIST Special Publication 800-56A, 2007 March
- [NIST SP800-57] National Institute of Standards and Technology (NIST), Recommendation for Key Management – Part 1: General, NIST Special Publication 800-57, 2007 March
- [NIST SP800-90] National Institute of Standards and Technology (NIST), Recommendation for Random Number Generation Using Deterministic Random Bit Generators, NIST Special Publication 800-90, 2007 March
- [NIST SP800-107] National Institute of Standards and Technology (NIST), Recommendation for Using Approved Hash Functions, NIST Special Publication 800-107, DRAFT
- [RFC 2119] IETF RFC 2119, Key words for use in RFC's to Indicate Requirement Levels, 1997
- [SA Core] TCG Storage Architecture Core Specification, Version 1.0, - IN PROGRESS, TBD
- [KMSS-1] TCG Storage Working Group Application Note 1, Version 1.0, Revision 0.39, 2007 Sep
- [K&R] Kernighan, Bryan and Ritchie, Dennis, C Programming Language, Prentice-Hall, 1988

3 Definitions, Abbreviations, and Conventions

3.1 Terms

- **Authenticate:** prove the integrity or identity of an entity
- **Authorize:** grant an authenticated entity access to a feature
- **Connect:** a process that authenticates a user and grants access to the Secure Volume
- **Constant:** a value or a set of values that do not change
- **Device:** supplier of services consumed by Host
- **ExternalHost:** a Host that is not a LocalHost and is capable of establishing a trusted session with the TPer, usually a network service or a locally attached peripheral; communication between an ExternalHost and the TPer is facilitated by LocalHost, but the LocalHost does not have knowledge of trusted session keys
- **FDE:** Full Disc Encryption
- **Host:** consumer of services supplied by Device; may be LocalHost or ExternalHost
- **Initializer:** a type of Cryptographic Officer as defined in [FIPS140-3]; a Trusted Authority with authorization to modify disc parameters, enroll, modify and delete other users, but without authorization to mount the Secure Volume
- **Invariant:** an assertion that is true over the lifetime of a specified object
- **IT:** Information Technology
- **LocalHost:** the MMC Host and its software application that is directly attached to an optical drive
- **Maker:** device manufacturer
- **MMC:** Multi-Media Commands standard from the ANSI / INCITS T10 committee[MMC-5]
- **Multi-factor authentication:** multiple methods are required to authenticate an entity
- **Owner:** legal owner of the Device, typically an IT department or an individual
- **[OSSC]:** Optical Security Subsystem Class, this document
- **Passcode:** a password, pass phrase, or PIN that is used to identify a user
- **Protected Storage Area (PSA):** a confidential area of the optical disc managed by a TPer.
- **Secure Volume:** the remaining address range after VolumeZero and the PSA have been reserved from the disc standard address range; the TPer presents the start of the Secure Volume as LBA 0 after a user is Connected
- **TCG disc:** a disc that has an initialized Protected Storage Area
- **Token: a** locally attached peripheral that is used by a LocalHost in multi-factor authentication
- **TPer:** Trusted Peripheral; the capabilities and behavior of the device that implement this [OSSC]
- **Trusted Authority:** an entity that performs mutual authentication with a TPer and establishes trusted session keys
- **User:** User Role as defined in [FIPS140-3]; an entity that is connecting or is connected to the disc, often a human
- **VolumeZero:** a clear-text volume that may be mounted before user connection to the Secure Volume; it has fixed size of 8192 MB at LBA[0..0FFFh] of the standard address range;
- **Well Known:** a published constant

3.2 Keywords

Key words are used to signify the requirements in the specification. The key words "**shall**", "**should**", "**may**," and "**optional**" are used in this document. These words are a subset of the RFC-2119 key words used by TCG, and have been chosen since they map to key words used in the MMC standard. These key words are to be interpreted as described in [RFC 2119].

Additionally, the following terms are used in this document to describe the requirement of particular [SA Core] features, including templates, tables, methods, and usages thereof.

- **Mandatory (M):** The feature **shall** be supported by the device in order to be compliant with this OSSC.
- **Optional (O):** The feature **may** be supported by the device.
- **Excluded (X):** The feature **shall not** be implemented by the device in order to be compliant with this OSSC.
- **TPer Private (T)**: The feature is private to and managed by the TPer; table / column/ object **shall not** be accessible.
- **Well Known (K):** The feature is Constant and **shall** be documented in this [OSSC]; table / column / object **shall not** be accessible.

Devices that implement SA Core features that are tagged Excluded (X) or Well Known (K) have additional security evaluation and conformance burdens.

Description and specification of document elements are declaratory and use the semantics of the Object Constraint Language [OCL] [Meyer]. In particular, invariants, preconditions, postconditions and descriptive assertions are used as suggested by [FIPS140-3]. This methodology results in creep from specification into design, but justification is claimed on the basis of Storage Workgroup deliverables. These include security profiles that have compliance, conformance, and certification aspects, and this methodology contributes to this objective.[SCS]

3.3 Bit and byte ordering

Endianness affects how processors load data wider than a byte into their registers for processing. Little endian processors assume the byte containing the least significant byte is located at the lowest memory address; big-endian processors assume the byte containing the most significant byte is located at the highest memory address.

Table 1 shows a little-endian 32-bit register (e.g. Intel x86).

Table 2 shows a big-endian 32-bit register with little-endian bit numbering (e.g. Motorola 68K).

Table 3 shows a big-endian 32-bit register with big-endian bit numbering (e.g. Power PC).

Table 1 — Little-endian 32-bit register (e.g. Intel x86)

When 32 bits are loaded from memory address 0, memory bytes go into register bytes as shown	**(MSB) 31:24**	**23:16**	**15:8**	**(LSB) 7:0**
	Byte 3	Byte 2	Byte 1	Byte 0

Table 2 — Big-endian 32-bit register (with little-endian bit numbering (e.g. Motorola 68K)

When 32 bits are loaded from memory address 0, memory bytes go into register bytes as shown	**(MSB) 31:24**	**23:16**	**15:8**	**(LSB) 7:0**
	Byte 0	Byte 1	Byte 2	Byte 3

Table 3 — Big-endian 32-bit register (with big-endian bit numbering (e.g. Power PC)

When 32 bits are loaded from memory address 0, memory bytes go into register bytes as shown	**(MSB) 0:7**	**8:15**	**16:23**	**(LSB) 24:31**
	Byte 0	Byte 1	Byte 2	Byte 3

TCG data structures are always big endian, meaning that when a field contains more than one byte, the byte containing the most significant bit is stored at the lowest address. MSB (most significant bit) and LSB (least significant bit) labels are provided for multi-byte quantities to serve as indications that software running on little-endian processors needs to reverse those bytes before processing the data.

3.4 Notation Conventions

The following operators[K&R] may be used in this [OSSC}:

- **x + y:** modular addition of two strings x and y
- **x** – **y**: modular subtraction of two strings x and y
- **x * y**: modular multiplication of two strings x and y
- **x** % **y**: modular remainder of two strings x and y
- **x | y**: bitwise OR of two strings x and y
- **x & y**: bitwise AND of two strings x and y
- **x ^ y**: bitwise exclusive-OR (XOR) of two strings x and y
- **~x:** bitwise inversion of x
- **sizeof(x):** the size in bytes of x
- **x < y**: less than
- **x <= y**: less than or equal to
- **x > y**: greater than
- **x >= y**: greater than or equal to
- **x = y:** equal to, '==', not assignment
- **x ≠ y:** not equal to, '!='
- **msb_n(x):** n most significant bits of x
- **lsb_n(x):** n least significant bits of x
- **$range_{n:m}$(x):** inclusive range of bits between bit n and bit m of x
- **x || y**: ordered concatenation of two strings x and y
- **pre(x)**: state of x when operation is entered
- **post(x)**: state of x when operation is exited

4 Overview (Informative)

4.1 Introduction

The Optical Security Subsystem Class [OSSC] is a framework for implementing Full Disk Encryption (FDE) for optical drives and discs. The primary capabilities specified by [OSSC] are:

- Authentication - Verification of identity
- Authorization - Verification of permissions for an authenticated entity
- Privacy/Confidentiality - Making secret that which is intended to be secret, and maintaining secret that which is intended to be secret.

According to [OSSC], users secure data with one or more pass-codes. A disc that is recorded according to [OSSC] may be shipped by non-secure carrier or even become lost, and the user may be confident that the data will not be exposed to unauthorized parties.

4.2 Optical Security Subsystem Class (OSSC)

OSSC security is managed by a set of tables and functions that operate on those tables. For a disc that has already been initialized by the OSSC, there are two types of tables:

- Tables stored on the disc,
- Volatile tables that are created for OSSC session management.

When a blank disc is loaded, the Host is responsible for determining the usage. The Host may choose to initialize the disc as an OSSC disc, or the Host may choose to use the disc in a different way. Drives may be configured to permit only secure recording according to [OSSC].

OSSC Security operations permit the Host to use the OSSC tables for the intended security purposes. The Host requests specific OSSC Security operations by using Security Protocol = 06h and Security Protocol Specific = 7007h in both the SECURITY PROTOCOL IN and SECURITY PROTOCOL OUT commands.

4.3 Cipher Schemes

[OSSC] specifies public key encryption for establishing secure communications channels between Host and Drive.

[OSSC] specifies the Advanced Encryption Standard (AES) for securing user data. AES-128 is mandatory. AES-256 is optional.

The SHA-256 Hash-only function is used for key derivation and random number generation.

The SHA-256 HMAC function is used is used for secure message integrity and entity authentication.

4.4 Disc Areas

4.4.1 Physical Volume

According to the profile for the medium, there is a PSN = D that is associated with LBA = 0. The Physical Volume is the sequence of sectors that begins at D and proceeds until the maximum capacity of the user data area. See Figure 1.

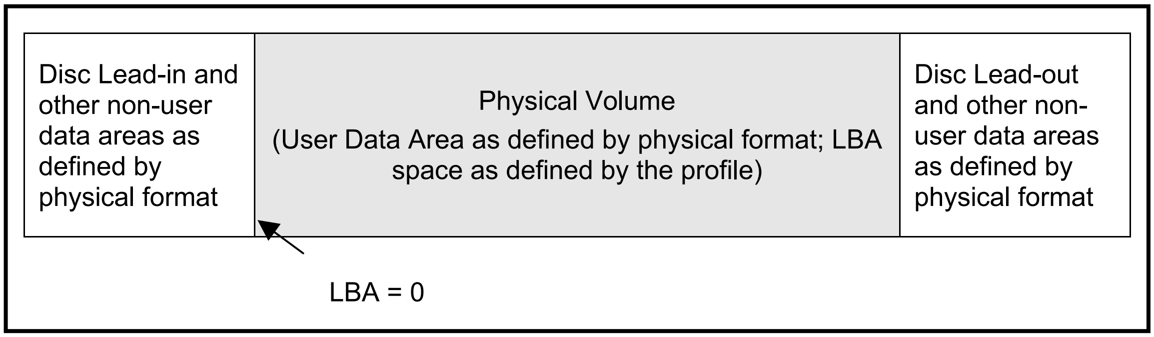


Figure 1 — Physical Volume

4.4.2 VolumeZero

In the OSSC Format, VolumeZero (Figure 2) begins at PSN = D and is allocated a length of 8 MB. Exact size is media type dependent; see [MMC-6]. VolumeZero is made available to the Host as a small, clear-text volume. One use for VolumeZero is to contain a read-only file system that may aid the Host in dealing with backward compatibility issues.

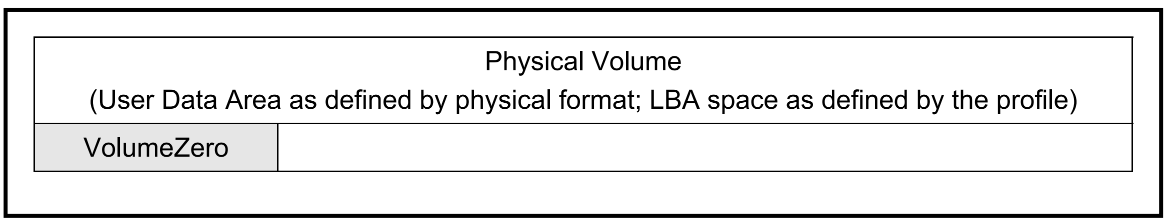


Figure 2 — VolumeZero

4.4.3 Protected Storage Area (PSA)

The [OSSC] requires a Protected Storage Area (PSA) – storage that is persistent, not included in the LBA space of the encrypted area, and not affected by Host partitioning. The PSA is defined in the disc's user data area in order to provide common security mechanisms over a wide range of optical media.

A PSA follows VolumeZero, however, the exact starting location and size are determined by the physical format of the media. When the media profile permits only sequential recording, additional areas may be taken for updating the PSA content.

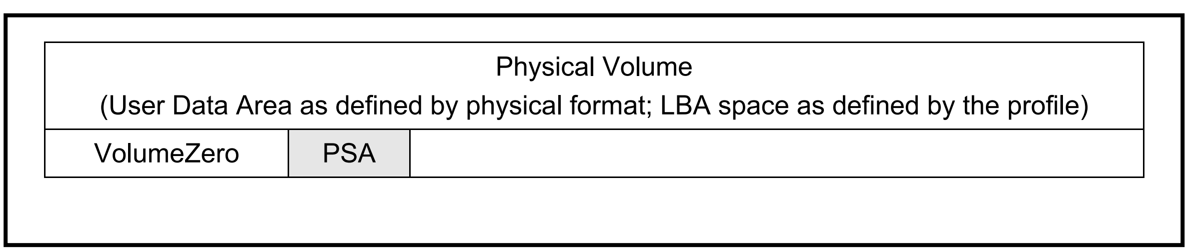


Figure 3 — General Location of the PSA

4.4.4 Secure Volume

As shown in Figure 4 the Secure Volume follows the PSA, however, the exact starting location and size are determined by the physical format of the media. When used according to the [OSSC], all user data sectors in the Secure Volume are encrypted.

Physical Volume (User Data Area as defined by physical format; LBA space as defined by the profile)		
VolumeZero	PSA	Secure Volume

Figure 4 — General Location of the Secure Volume

The OSSC overhead areas typically require less than 1% of the capacity of the Physical Volume. Consequently, the Secure Volume contains > 99% of the Physical Volume.

4.5 OSSC TPer Users

There are 2 OSSC user types:

1. Initializer

   The Initializer is an administrative cryptographic officer. The Initializer user is the only authority permitted to change user information. Consequently, the Initializer may enroll common users and the Initializer may modify or delete any user record. However, the Initializer is not permitted access to the Secure Volume.

2. Common

   A Common user is not permitted to modify any user's record - including its own. A Common user may provide LBA access only to the Secure Volume.

4.6 Templates, Tables, and Methods

A Template as defined by [SA Core] is a set of definitions for tables and methods. When tables are allocated according to a Template, the result is an "issued" Security Provider (SP).

Several templates are defined by [SA Core] and an additional template is defined for specific use by Optical Storage devices. The templates that are potentially applicable to OSSC TPers are shown in Table 4.

Table 4 — OSSC TPer Usage of [SA Core] Defined Templates

Template	Description
Admin	The Admin template permits issuance of SPs and maintains information about the TPer
Base	The Base template provides functionality for all SPs
Locking	The Locking template defines mechanisms for access control to user data, including controlling media encryption, user data encryption key management, and Read/Write lock state.
Optical	The Optical template provides the functionality that is unique to the OSSC TPer

Since the OSSC TPer contains only the power-on defined SP, the OSSC TPer and the Optical SP may be viewed to be a single entity. The Base and Locking templates are not implemented by OSSC TPers.

4.6.1 Optical Template Tables

The Optical template is used to define the Optical Security Provider (OSP). Some of the OSP tables are written into the PSA while others are constructed from Drive and general disc information. The part of the OSP that is written into the PSA is the OSP Basis (OSPB). Critical Security Parameters in the OSPB are encrypted.

Table 5 shows the tables in the Optical template

Table 5 — Optical template tables

Table	Description
Drive	The TPer constructs this table to provide information about drive capabilities, drive state, and includes personalization information.
Anchor	Anchor enables interchange and indexes all tables that are on-disc.
Disc	Disc is a descriptor that includes information that apply to the entire disc.
C_User	Each row in C_User represents a single user. Each user is associated with an identifier (Name), a Pass-code and, optionally, multiple authentication factors. A row in C_User is known as a user record.
SessionMap	The TPer uses this table to manage PSA updates on disc formats that require a track/session recording methodology.

4.6.2 Methods

Methods are functions in the Drive that may be executed upon requests from the Host. [SA Core] defined methods used by the optical TPer are shown in Table 6.

Table 6 — Methods used by Optical TPers that are defined by [SA Core]

Method	Description
Get	The Host may invoke the Get on any table row in order to be given a copy of its content. When table content is considered private, the Host **shall** be denied the data.
Set	The Host may invoke the Set on any table row in order to specify its content. When table content is considered private or invariant, the Host **shall** be denied the ability to change the information.
StartSession	This method is used to start a session with the TPer.
SyncSession	This method is used to synchronize session start with the Host.
StartTrustedSession	This method is used to continue session setup to the trusted state.
SyncTrustedSession	This method is used to synchronize trusted state with the Host.
CloseSession	When a Host application has completed its work with the Drive, the secure session should be closed.

Managing the Optical TPer requires methods that are defined in this [OSSC] specifically for use by an Optical TPer. These are listed in Table 7, Table 8, and Table 9.

Table 7 — Secure Firmware Upgrade

Methods	Purpose
FWBegin	Enter FIPS approved firmware download mode
FWEnd	Exit FIPS approved firmware download mode

Table 8 — Optical Disc Management

Methods	Purpose
MountSV	If Secure Volume is not mounted, dismount current volume and MountSV.

Table 9 — User Management

Methods	Purpose
SetPointer	Focus the attention of the OSP to a specific C_User row
EnrollBegin	Begin enrollment of user of Secure Volume
EnrollEnd	Complete enrollment of user of Secure Volume
ConnectBegin	Begin connection of user to the Secure Volume
ConnectEnd	Complete connection of a user to the Secure Volume
Factor	Authentication factor
EraseUser	Remove a user from the list of users that may connect to the Secure Volume
Disconnect	Disconnect a connected user

5 Features and Capabilities

5.1 Interface Communications Protocol

[OSSC] compliant devices implement the commands: SECURITY PROTOCOL IN (A2h) and SECURITY PROTOCOL OUT (B5h) defined in the SCSI standard, [SPC-4]. In order to generalize for other interfaces and command sets, [SA Core] refers to these commands respectively as IF-RECV and IF-SEND.

[MMC-6] requires that the optical drive report the Trusted Computing Feature Descriptor in the response data of the GET CONFIGURATION (46h) command.

When OSSC behavior is not current, [OSSC] devices behave exactly as legacy, [MMC-5] devices. There is no new interface error behavior expected of OSSC devices; for example, READ (10) returns cipher-text when a user is not connected. As a general rule, security errors are contained within the security protocol and are not elevated to interface errors.

5.2 AES Block Cipher Algorithm

Symmetric cryptographic algorithms are based on the Advanced Encryption Standard (AES) block cipher algorithm as specified in [FIPS 197]. Both AES-128 and AES-256 are permitted; AES-128 is Mandatory.

For encryption of the Secure Volume, the AES cipher is used in the Cipher Block Chaining (CBC) mode as specified in [NIST SP800-38A]. Message length is one sector (2048 bytes). Each message has a statistically unique Initialization Vector (IV). The calculation for the IV is:

IV = AES_encrypt(msb_{96}(DiscIVbase) || LBA, FDEkey), where

- DiscIVbase is a 128-bit random number assigned when the PSA is initialized.
- LBA is the logical block address as defined by standard disc formats before the address remapping of this [OSSC] is applied (see “Physical Disc Address Space” as shown in Figure 1 through Figure 4).
- FDEkey is the Full Disc Encryption key assigned when the PSA is initialized.

Secure message confidentiality uses AES-128/CBC. The encryption key is a byproduct of establishing the trusted session; the IV is a random number included in the message. Secure messaging is permitted only after a trusted session has been established.

PSA confidentiality uses AES-128/CBC. The encryption key is a byproduct of §5.3.2.

5.2.1 SHA-256 Hash-only Function

[FIPS180-3] SHA-256 is used for key derivation and random number generation.

5.2.2 SHA-256 HMAC Function

[FIPS198-1] SHA-256 HMAC is used for secure message integrity and entity authentication.

5.2.3 Key Derivation and Entity Authentication

DeriveKey( ) is constructed with [NIST SP800-56A] Concatenation Key Derivation Function (Approved Alternative 1) where:

- Z is the pass-code (provided by a Host in the method Factor).
- KeyDataLength is either 128 bits or 256 bits, based upon the selected AES encryption.
- The hash function is SHA-256. Consequently, HashLen is 256.
- OtherInfo is the Salt for key derivation. Each field within OtherInfo is defined by the Salt supplier.

The result of DeriveKey() is a Derived Key (DK) that represents an authentication factor.

MakeEAC( ) and CheckEAC( ) are constructed with [FIPS198-1] using SHA-256 HMAC. They operate on **E**ntity **A**uthentication **C**odes (EAC) that are used to check the integrity of user authentication factors.

The Salt and HMAC key have one of two possible sources:

- TPers shall contain internal Salt and HMAC (see Appendix B). The TPer internal parameters are secured within the Optical Disc Drive and are used by default. The TPer internal Salt and HMAC key are never written to a disc or exposed to the host.
- LocalHost may override the use of the internal TPer parameters by supplying its own parameters to the Disc table (see 7.1.2).

The use of the TPer internal parameters provides the higher level of security, while use of LocalHost supplied parameters allows the possibility to use legacy (non-TCG) optical drives and a special software application for reading discs secured according to this SSC.

5.2.4 Random Number Generation

Random / Pseudorandom number generation is in accordance with [NIST SP800-90].

5.3 Authentication

5.3.1 Host Authentication

TPer clients, including LocalHost applications, authentication factors, and remote hosts participate in a public key, mutual authentication protocol with the TPer. The [SA Core] exchange method is used for client authentication; exchange authorities are listed in §5.4, Exchange. The purpose of Host authentication is to establish trust between the two authenticating parties and to establish symmetric keys for secure messaging.

5.3.2 User Authentication

User authentication is performed only within a trusted session that has been established with Host authentication. Secure messaging is used to protect the confidentiality of user credentials.

The TPer maintains a unique record for each user; the user record is constructed with authenticated encryption. Each user record contains:

- A clear-text user name for mapping the user record to the user.
- An Entity Authentication Code (EAC) that validates user authentication factors.
- A Key Encrypting Key (KEK, the PSAkey) that encrypts the sensitive PSA parameters.

The sequence of events during enrollment/connection are managed by the local Host on behalf of the user:

- LocalHost establishes a trusted session with the TPer on the user's behalf.
- LocalHost locates the associated user record according to the user's name.
- Via LocalHost and potentially ExternalHosts the user provides a Pass-code and optionally, multiple authentication factors to connect to a disc.
- The Drive generates a Derived Key (DK) for each authentication factor with DeriveKey( ), and each DK is combined into a DK_accumulator.
- Next, either MakeEAC( ) or CheckEAC( ) is executed to check the integrity of the accumulated authentication factors. When a user is being enrolled on the disc, the MakeEAC() algorithm is executed, the resulting EAC is inserted into the user record.
- If a user is being enrolled, DK_accumulator encrypts sensitive parameters in the user's record. If a user is being connected to the disc, DK_accumulator is used to decrypt sensitive parameters in the user record.

5.4 Exchange

Public key exchange authorities are supported in the Optical TPer and its clients:

- **TPer Exchange Authority** - Optical TPer's require a certificate to participate in mutual authentication procedure with client software and authentication factors.
- **LocalHost Exchange Authority** - LocalHost software participates in a mutual authentication procedure with the TPer before authorization is granted to confidential TPer features.

- **ExternalHost Exchange Authority** - Authentication factors and remote hosts participate in a mutual authentication procedure with the TPer before authorization is granted to confidential TPer features.

5.5 TPer Requirements

An optical TPer has capabilities that permit optical storage to support and conform to the access control policies of users and organizations. There are four themes that strongly influence the architecture of the optical TPer:

- Media Interchange
- Limited resources - including the PSA
- Read only (ROM), write once (R), and limited rewritable (RW) media
- Single process, synchronous MMC architecture

Optical tables are initially formed with device firmware at power-on. When a TCG disc is inserted, on-disc tables update the default tables. Optical tables are written to a protected storage area on the disc. All OSSC optical drives must be able to locate the on-disc optical tables, and semantically read the tables into the optical TPer. Because any drive may be able to read an OSSC disc, the PSA is partially encrypted. The encryption key for the optical tables is derived from external Pass-codes not stored on the disc. Without appropriate credentials, the on-disc optical tables cannot be decrypted and the data on the disc remains confidential.

Since limited resource is a concern, only one SP is mandatory, the Admin SP. The Admin SP is formed by device firmware. Many of the [SA Core] defined Admin SP tables are optional in this TPer. Convention is chosen over discovery and versatility: when table format and content are fixed, they are considered well known and existence within the Tper is not required. These decisions minimize the TPer footprint in the optical drive.

Historically, optical drives have been controlled by a single process using a synchronous command/response protocol. Streaming multimedia and write once recording works best with a single process/synchronous model. This [OSSC] limits the [SA Core] defined protocol stack and behavior so that the requirements imposed by this [OSSC] are compatible with MMC and the device architectures that support optical use cases.

5.6 Use Cases

Use cases encapsulate requirements to guide the development process. The Optical SSC supports five use cases:

0) Computing environment is personal/organizational desktops/notebooks
1) Individual user data archival
2) Organizational data distribution
3) Distribution of information under access control, e.g. electronic health records
4) Secure network endpoint, e.g. disaster and emergency response

5.7 Threat Model

The Optical SSC addresses the data-at-rest problem of lost or stolen optical discs; the following attacks are considered in the TPer and system components and their interfaces:

0) Missing disc: an adversary gains possession of disc
1) Data breach: an adversary gains access to sensitive data that is supposed to be kept secret at entities that are external to the TPer
2) Eavesdropping: an adversary snoops communications to learn useful information
3) Replay: an adversary records messages that were sent in past communications and re-sends them at a later time
4) Man-in-the-middle: an adversary intercepts the messages sent between entities and replaces them with its own messages

5) Password-guessing attacks: an adversary repeatedly picks a password from a dictionary and tries to use it in order to impersonate the user
6) VolumeZero attacks: an adversary causes mischief with the clear-text file system that is mounted before connection to the secure volume

6 Communications

6.1 Command Interface

If a command set supports TCG devices then that command set **shall** provide two commands:

1. The IF-SEND command permits a Host to send TCG method execution requests.
2. The IF-RECV command permits a Host to receive the response associated with each TCG method execution.

In the case of MMC optical devices the only command set supported is SCSI. The typical physical interface is ATAPI, either parallel or serial. The SCSI command set provides the SECURE PROTOCOL OUT command as the representative for IF-SEND and the SECURE PROTOCOL IN command as the representative for IF-RECV.

6.2 Common Command Structures (CCS) protocol

TCG devices that use the Common Command Structures (CCS) **shall** implement only Protocol ID 6 for IF-SEND and IF-RECV. When the Protocol ID is 6, the Protocol Specific parameter **shall** be ComID.

OSSC devices use only ComID 7007h for Local Host connections. In the case where N-factor authentication is used (where N > 1), local hosts may provide additional factors and external hosts use ComID 7008h in order to provide additional authentication factors over a secure channel. Regardless of ComID, OSSC devices use only the Common Command Structure (CCS) protocol.

Only TPer method calls are sent for ComIDs 7007h and 7008h with the IF-SEND command. Only TPer method call responses are returned for ComIDs 7007h and 7008h with the IF-RECV command.

6.3 Discovery

Clients ‘discover’ features and capabilities of the TPer. There is one discovery level specified by [MMC-6], one specified by [SPC-3], and 4 discovery levels identified by [SA Core].

6.3.1 MMC Discovery (Level -2)

[MMC-6] describes the operation of the GET CONFIGURATION command. This command returns a list of Profile Descriptors and Feature Descriptors for the purpose of reporting capabilities supported by the Drive. [MMC-6] includes the OSSC Feature. When the OSSC Feature is present, the Drive is capable of executing the SECURITY PROTOCOL IN command, the SECURITY PROTOCOL OUT command, recognizing OSSC initialized discs, authenticating users, and connecting to the Secure Volume. For details, see [MMC-6].

6.3.2 SPC Discovery Level (Level -1)

OSSC devices **shall** support the Security protocol information capability of the SECURITY PROTOCOL IN command (Security Protocol = 00h and Protocol Specific = 0000h). See [SPC-4].

6.3.3 SSC Capabilities (Level 0)

Level 0 discovery provides a Host with information about TCG device capabilities. Level 0 information is requested by sending the SECURITY PROTOCOL IN command with Security Protocol = 06h and Protocol Specific = 0001h. To receive all potential Level 0 data for a device that conforms to this [OSSC], an Allocation Length **shall** be at least 24.

If Level 0 discovery is performed for any other Protocol ID, the TPer **shall** return 4 bytes of zero.

6.3.3.1 Level 0 Returned Data

The Optical TPer returns 24 bytes of fixed level 0 discovery data as shown in Table 10.

Table 10 — Level 0 returned data

Byte \ Bit	7	6	5	4	3	2	1	0
0 - 7	Level 0 Discovery header							
8 - (N-1)	Feature Descriptors							

6.3.3.2 Level 0 Discovery Header

The Level 0 discovery header for Protocol ID 6 (Table 11) **shall** precede all other Level discovery data.

Table 11 — Level 0 Discovery Header

Byte \ Bit	7	6	5	4	3	2	1	0
0	(MSB)	Length of Parameter Data						
3								(LSB)
4	(MSB)	Data Structure Version						
7								(LSB)

Length of parameter data is the total number of bytes that are valid in the level 0 discovery data. The length of parameter data field does not include itself. Devices that conform to this [OSSC] **shall** set this field to 00000014h (20d).

The Data Structure Version Number describes the format of the level 0 discovery header returned. The value **shall** be set to 00000001h.

6.3.3.3 Feature Descriptor - Generic Format

A feature is a set of capabilities that may be implemented in a TPer. A Host may discover the capabilities and properties of a TPer by examining its feature descriptors. Features implemented by a TPer **shall** be indicated by the presence of a feature descriptor.

Feature descriptors (Table 12) **shall** be returned in the LEVEL 0 DISCOVERY response data in order of increasing feature code values. Features that are not implemented **shall not** be returned.

Table 12 — Generic Feature Descriptor

Byte \ Bit	7	6	5	4	3	2	1	0
0	(MSB)	Feature Code						
1								(LSB)
2	Version				Reserved			
3	Additional Length = N							
4 - (N+3)	Additional feature data							

The Feature Code field **shall** identify a feature implemented by the TPer.

The Version field describes the format of the data returned. Future versions of a feature **shall** be backward compatible; incompatible changes **shall** be included in a different feature.

The Additional Length field indicates the length of the number of bytes of Feature Dependent Data that follow this field. This field **shall** be an integral multiple of 4.

6.3.3.4 Optical Feature

The presence of the Optical Feature, indicates that the TPer is an optical TPer that is consistent with this [OSSC]. If the Optical Feature Descriptor (Table 13) is present, the TPer supports FDE on the optical disc's Secure Volume.

Table 13 — Optical Feature Descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)	Feature Code (0050h)						
1								(LSB)
2	Version = 0000b				Reserved			
3	Additional Length = 0Ch							
4	(MSB	OpticalTPerVersion						
5								(LSB)
6	Reserved							
7	Reserved							
8	(MSB)	LocalHost ComID = 7007h						
9								(LSB)
10	(MSB)	ExternalHost ComID = 7008h						
11								(LSB)
12	Reserved							
13	Reserved							
14	Reserved							
15	Reserved							

The Feature Code field **shall** be set to 0050h.

The Version field **shall** be set to 0h.

The Length field **shall** be set to 12(0Ch).

The OpticalTPerVersion field **shall** be set to 0001h.

The LocalHost ComID **shall** be set to 7007h.

The ExternalHost ComID **shall** be set to 7008h.

6.3.4 SSC Communications (Level 1)

Level 1 discovery provides the Host with communications information according to [SA Core]. The information is returned as a response to the Properties method call. The OSSC TPer does not utilize any [SA Core] defined communications, and thus returns no data in the response to the Properties method call.

6.3.5 SSC General Tables and Methods (Level 2)

Level 2 discovery is associated with collecting information about the TPer by using the Get method to obtain copies of the SP and TPerInfo tables.

6.3.6 SSC Specific Tables and Methods (Level 3)

Level 3 discovery is associated with collecting information about the TPer by using the Get method to obtain copies of the Drive and Disc tables.

6.4 Communication Behavior

6.4.1 ComID and Session Management

An OSSC TPer **shall** respond only to ComID's 7007h and 7008h. ComID 7007h is used exclusively for LocalHost communications. ComID 7008h is used only for authentication factors from ExternalHosts.

An OSSC TPer **shall** allow only one active session per ComID.

An OSSC TPer **shall** automatically close an active session when StartSession[ ] is invoked on the same ComID.

6.4.2 Command Processing

The TPer executes a method request from a Host only when requested via the SECURITY PROTOCOL OUT command.

The TPer returns a method response only as a response to the SECURITY PROTOCOL IN command.

- When a session is open, the OSSC device **shall** ignore any method request when an invalid Session ID is detected. See 8.4.2.
- When a session is open and the CCS SeqNumber field is not valid (i.e. SeqNumber is lower than or equal to the SeqNumber in a previously acted upon method request) the OSSC device **shall** ignore the method request and close the open session. See 8.4.2.

When a session is open and a method request is made with SeqNumber set to zero, the OSSC device **shall** ignore the method request and close the open session.

- When a session is open and disc eject is requested, the OSSC device **shall** close the open session.
- An OSSC device **shall** report the NOT_AUTHORIZED error for any method call that requests access to:
- Excluded (X) tables and methods,
- TPer Private (T) tables, methods and objects, and
- Well Known (K) tables.

7 Optical Template

The Optical template contains cryptographically protected disc keys, User information and an application specific access control mechanism, and is partially contained within the disc PSA.

7.1 Optical Template Tables

7.1.1 Drive table (M)

Drive table is formed by TPer firmware and is not written to the disc. It publishes the optical drive capabilities, drive state, and includes a TrustedMode personalization that requires user connection before writing is authorized. Table 14 defines individual Drive table cell content.

Table 14 — Drive table Definition

Column	Type	R/W	Description
CCS UID	uinteger_2	R	CCS UID: 0080h
DriveAES	uinteger_1	R	Drive AES encryption capability: When DriveAES = 1, the drive is AES-128 capable. When DriveAES = 2, the drive is AES-256 capable. When DriveAES = 3, the drive is AES-128 and AES-256 capable.
TrustedMode	uinteger_1	W1R	Forced encryption personalization. When TrustedMode = 0, recording non-OSSC format discs is permitted. When TrustedMode = 1, User **shall** be connected before recording is authorized.

Notes:
1. R/W = R means the cell is read-only
2. R/W = W1R means: If the cell value is zero, SET is permitted. If the cell value is non-zero, the cell is read-only
3. TrustedMode = 0 is a bypass capability and is should be considered an unapproved FIPS mode (See [FIPS140-3]).

The MMC-6 defined OSSC Feature Descriptor contains a Mandatory Encryption bit, ME. This creates two behaviors for the TrustedMode value:

1. When ME is set to zero, TrustedMode **shall** be initialized to zero at power-on time. Drive.Set may be used to set TrustedMode to one. If TrustedMode is set to one, Drive.Set **shall not** be permitted to set TrustedMode to zero.
2. When ME is set to one, TrustedMode **shall** be initialized to one at power-on time. Drive.Set **shall not** be permitted to set TrustedMode to zero.

7.1.2 Disc table (M)

Disc table contains information about the disc. Disc table is initially formed by the TPer using default values. After connection to an OSSC disc, the TPer overwrites the default values with the on-disc structures. When a disc is removed, the TPer reverts Disc table to its default values.

Disc table or a copy of Disc table **shall** be located in the first 192 bytes of the first block after an Anchor. Disc table is a write-once table, it **shall not** be changed after it is initially written and subsequent instances of Disc table **shall** be exact copies of the initial Disc table.

Disc table includes a cell (HostMAC) for application specific metadata. For example a Host may wish to store a digital signature of the VolumeZero content.

The TPer possesses internal parameters: Salt required by the DeriveKey() function and an HMAC key required by the MakeEAC() and CheckEAC() functions (see Appendix B). For less secure applications, the LocalHost may override the use of the internal TPer parameters by supplying its own parameters via the disc.set method as follows:

1. The Boolean HSP_valid is set to 1.
2. Bytes 0 through 31 of the HSP cell are set to specify the DeriveKey() salt.
3. Bytes 32 through 63 of the HSP cell are set to specify the HMAC key used by the MakeEAC() and CheckEAC() functions.

Table 15 — Disc Table Definition

Column	Type	Encrypted by PSAkey	R/W	Description
CCS UID	uinteger_2	F		CCS UID: 0081h
CryptoType	uinteger_1	F	W1R	Selects FDE Cryptotype. Initialized to 00h. = 0 means not selected = 1 means use AES-128 (default) = 2 means use AES-256
HSP_valid	Uinteger_1	F	W1R	Selects Host parameters for key derivation and entity authentication = 0 means use TPer parameters (default) = 1 means use Host supplied parameters
HSP	Bytes{64}	F	W1R	Host Supplied Parameters Salt = bytes 0 –31 HMAC key = bytes 32 - 63
DiscUniqueID	bytes{16}	F	R	Unique Disc ID for Host
HostMac	bytes{32}	F	W1R	Host managed MAC
Reserved	Bytes{28}	F	H	Reserved (zero filled)
FDEkey	bytes{32}	T	H	FDEkey encrypted
DiscIVbase	bytes{16}	T	H	Nonce as a base IV for FDE

Notes:
1. R/W = R means the cell is read-only
2. R/W = W1R means: Once a value has been SET, the cell becomes read-only
3. R/W = H means that the cell is hidden and is neither readable nor writable

7.1.3 C_User (M)

Each row in C_User represents a single user. Each C_User row contains sufficient information to identify the user and permit secure connection. The format of a C_User row is shown in Table 16.

Table 16 — C_User Row Definition

Column	Type	IsEncrypted	R/W	Description
RowX	uinteger_2	F	R	Row Index
UserName	bytes{64}	F	W1R	Array of bytes that is not all zeros.
IsValid	uinteger_1	F	R	Row validity: = 0 means this row is empty = 1 means this row is valid
Reserved	bytes{29}	F		Reserved (zero filled)
Nfactors	uinteger_1	F	H	Number of authentication factors; private to TPer
Reserved	bytes{95}	F	H	Reserved (zero filled) for confidential use
PSAkey	bytes{32}	T	H	PSA key encrypted with DeriveKey() result. Key size is specified by Disc.CryptoType. If Disc.CryptoType = AES-128, then 16 most significant bytes are set to zeroes.
EAC	bytes{32}	T	H	Entity Authentication Code used to validate user authentication factors

Notes:
1. R/W = R means the cell is read-only
2. R/W = W1R means: Once a non-zero value has been SET, the cell becomes read-only
3. R/W = H means that the cell is hidden and is neither readable nor writable

The first enrolled user is the Initializer user. All other users are enrolled as common users. C_User UIDs are Well Known and thus occupy no space in a C_User row.

7.1.4 Anchor (T)

7.1.4.1 Structure and Content

Anchor table identifies the locations of all OSP tables that are stored on-disc. Anchor table UID is null and **shall not** be accessible.

Table 17 — Anchor Table Row Definition

Column	IsOrdered	Type	Description
RowType	yes	uinteger_1	Table identifier.
Reserved		bytes{3}	Reserved
Name		bytes{16}	Table name; character set = UTF-8; right padded with zeroes
PSARSA		uinteger_4	PSA Relative Sector Address of the sector that contains the named table
Offset		uinteger_2	Byte offset from start of the sector that contains the named table
SeqOrSeg		uinteger_2	When RowType = 04h (C_User table segment), this field contains the C_User table segment number. When RowType ≠ 04h, this field is a sequence number of the table where a sequence number represents a monotonically increasing version number.
Size		uinteger_4	Size of table identified by RowType

The Anchor table row numbering is specified in Table 18.

Table 18 — RowType Numbers

RowType	Table
01h	Anchor table
02h	Disc table
03h	SessionMap table
04h	C_User table segment
00h	Uninitialized Anchor table row

7.1.4.2 Anchor table Row Ordering

The Anchor table row numbering has a number of ordering requirements:

1. The Anchor table row **shall** appear first.
2. The Disc table row **shall** appear second.
3. The SessionMap table row **shall** appear third.
4. The first C_User table segment row **shall** appear fourth.
5. If there is more than one C_User table segment row, all C_User table segment rows **shall** appear contiguously in the Anchor table.
6. All C_User table segment rows **shall** appear in increasing segment number order.
7. Each row that is not in use **shall** be zero filled.

The RowType and Name cells of the first three rows of an Anchor are mandatory and constant. This invariant allows Anchor table to be used as a signature to determine OSSC discs.

The initial Anchor table format is shown in Table 19.

Table 19 — Initial Anchor table

RowType	Name	PSARSA	Offset	SegOrSeq	Size	Comment
01h	Anchor	0	0	1	2048	this table
02h	Disc	1	0	1	192	Disc descriptor; initialized exactly once; may be copied many times
03h	SessionMap	1	192	1	1024	PSA descriptor of write once physical volume parameters
04h	C_User	2	0	1	N	1st segment of C_User rows; N = 14*2048 for DVD or N = 30*2048 for HD-DVD and Blu-ray
0	0	0	0	0	0	null rows to complete sector
more null rows …						
0	0	0	0	0	0	null rows to complete sector

Anchor table UID = null and **shall not** be accessible to Hosts.

Rows of RowType = 00h (NULL) **shall** be ignored by a TPer.

Disc.SegOrSeq = 1 because Disc table is a write once table. Subsequent updates to the PSA **shall** have exact copies of the initial Disc table.

A TPer **shall** record the initial Anchor at a Well Known starting location. Given a specific media type, the starting location is Well Known and fixed.

Construction of the PSA and indexing by the Anchor table is provided in §7.2, Protected Storage Area Construction. [MMC-6] specifies how the Anchor table and the PSA are updated and located with the various optical recording technologies, including write once and limited rewrite.

In this [OSSC], Anchor table **shall** be contained entirely within one sector.

The minimum number of C_User rows **shall** be the number of sectors in a single writable unit minus two (2), where a single writable unit is defined by disc format (32K for DVD and 64K for HD-DVD and Blu-ray).

7.1.5 SessionMap (T)

The TPer uses this table to manage PSA updates on physical volumes that implement recording using the track/session model. Its entry remains in the Anchor table, but it is not present on discs that implement recording using the random writable model. Write Once volumes are organized with sessions and tracks. Each recording technology uses its own synonyms for session and track, but the semantics are the same. See [MMC-6] for details. SessionMap UID = null and **shall not** be accessible. Table 20 shows the row format of the SessionMap table. SessionMap is limited to 256 rows.

Table 20 — SessionMap table row Definition

Column	Type	Description
TrackStart	uinteger_2	PSA start track (Logical Track number)
SessionStart	uinteger_2	PSA start session (Session number)

7.2 Protected Storage Area Construction

This section specifies how a TPer constructs the internal structure of the PSA. It is concerned only with the internals of the PSA; location of the PSA on a disc is specified in [MMC-6]. The optical PSA is more complex than other storage technologies because of the interchange requirement and the nature of the recording technologies, write once and limited rewrite.

The PSA is the address range where optical tables are on-disc. Within the PSA are a collection of tables, and the set of the most recent Anchor tables and its referenced tables is called the **O**ptical **S**ecurity **P**rovider **B**asis (OSPB).

Two types of OSPB's **may** be constructed depending on device capabilities, a Singular OSPB for resource limited devices and a Linked OSPB for organizational applications where a large number of users is required. All devices **shall** be able to connect all users enrolled on a Linked OSPB, but resource limited devices **may not** be capable of originating a Linked OSPB.

Both OSPB types limit the number of users that **may** be enrolled on-disc. However, organizations **may** increase the number of users that **may** connect to the disc by treating an on-disc User as a class and augment the class with external authentication methods.

The Anchor table is a general and versatile data structure, and receiving TPer's **shall** locate tables within the OSPB as given in the Anchor table. Originating TPer's **shall not** create an incorrect Anchor table. There **shall** be a one-to-one association between a table referenced in an Anchor table row and the actual table within the OSPB. To ensure a correct OSPB, the two following rules **shall** be followed:

- Two Anchor table rows **shall not** reference the same location within the OSBP
- All locations within an OSPB **shall** be referenced by an Anchor table row.

The most complex PSA activity performed by a TPer is management of C_User tables and rows. In this [OSSC], C_User segments (SegOrSeq) are managed with the writable unit of the physical disc standard, 16 sectors (32 KB) for DVD and 32 sectors (64KB) for HD-DVD and Blu-ray. Furthermore once a writable unit is written on-disc, modification of any table within the writable unit **shall** cause the entire writable unit to be rewritten at the convenience of the TPer.

[MMC-5] specifies additional details for management issues that are related to recording technology.

7.2.1 Initialization

Initialization refers to the enrollment of the initial user (the Initializer) by LocalHost. The initialization procedure is the same for both OSBP types and is shown in the following figure. In a Singular DVD Anchor table, PSARSA sectors greater than 0000 000Fh do not exist.

If the Initializer is the only user specified in the C_User table, Figure 5 shows how the OSPB appears after PSA recording.

Anchor					
RowType	**Name**	**PSARSA**	**Offset**	**SegOrSeq**	**Size**
01h	Anchor	0	0	1	0000 0800h
02h	Disc	1	0	1	0000 0040h
03h	SessionMap	1	192	1	0000 0400h
04h	C_User	2	0	1	N
00h	0	0	0	0	0
more null rows ...					
00h	0	0	0	0	0

<table>
<tr><td colspan="2">PSA map</td><td></td><td></td></tr>
<tr><td>Sector</td><td>Table</td><td></td><td></td></tr>
<tr><td>0</td><td>Anchor</td><td></td><td></td></tr>
<tr><td rowspan="2">1</td><td>Disc</td><td></td><td></td></tr>
<tr><td>SessionMap</td><td colspan="2">C_User Column</td></tr>
<tr><td rowspan="5">2</td><td rowspan="5">C_User</td><td>Row UID LSB</td><td>IsValid</td></tr>
<tr><td>00h</td><td>2</td></tr>
<tr><td>01h</td><td>0</td></tr>
<tr><td>...</td><td>0</td></tr>
<tr><td>07h</td><td>0</td></tr>
<tr><td colspan="4">more DVD, HD DVD, Blu-ray sectors...</td></tr>
<tr><td rowspan="3">0Fh</td><td rowspan="3">C_User</td><td>68h</td><td>0</td></tr>
<tr><td>...</td><td>0</td></tr>
<tr><td>6Fh</td><td>0</td></tr>
<tr><td colspan="4">more HD DVD, Blu-ray sectors...</td></tr>
<tr><td rowspan="3">1Fh</td><td rowspan="3">C_User</td><td>E8h</td><td>0</td></tr>
<tr><td>...</td><td>0</td></tr>
<tr><td>EFh</td><td>0</td></tr>
</table>

Figure 5 — Initial OSPB in PSA

7.2.2 Singular OSPB

A TPer **may** constrain the size of an Anchor table and its referenced tables to one writable unit, 32KB for DVD and 64KB for HD-DVD and Blu-ray. This limits the maximum number of on-disc users to 112 for DVD and 240 for HD-DVD and Blu-ray. The Anchor table and all its referenced tables are rewritten with each modification. All relevant information is contained in the most recently written writable unit of the PSA. See [MMC-6] for details on where and how the PSA is recorded on the various disc standards. Attempting to add more users than there is space for will result in an error with a status code of INSUFFICIENT_SPACE (Section 5.1.3 of the Core Specification).

7.2.3 Linked OSPB

A TPer that implements the **Optional** Linked OSPB should have a buffer that holds a minimum two writable units, 64KB for DVD and 128KB for HD-DVD and Blu-ray. The maximum number of on-disc users is limited by the number of C_User rows in Anchor table, 7,680 users for DVD and 15,360 users for HD-DVD and Blu-ray.

Rather than limit the Anchor table and its referenced tables to a single writable unit, the Anchor table **may** reference tables written in writable units other than the one that contains the Anchor table. The referenced writable units may have been written earlier than the Anchor table.

TPer's **shall** write OSPB's sequentially until of the end of the PSA. When a new OSPB is required, a new Anchor table **shall** be written in the next available rewritable unit; OSBP modifications **shall** be written in subsequent, sequential recordable units. With random rewritable recording, OSPB's **shall** wrap within the PSA as defined in [MMC-6].

7.2.3.1 C_User{ } Spanning Multiple Writable Units (Informative)

As users are enrolled on a disc, new writable units are allocated to enroll additional users. In the following figure, the SegOrSeq of the Anchor row has incremented to reflect a new version of the Anchor table at PSARSA 10h, and two new C_User rows have been added at PSARSA 20h and 30h. C_User SegOrSeq 1 is also rewritten because it is contained in the same writable unit of the Anchor table. The same applies to Disc table and SessionMap.

Table 21 — New Users Enrolled on a DVD Linked OSPB

Anchor					
RowType	**Name**	**PSARSA**	**Offset**	**SegOrSeq**	**Size**
01h	Anchor	0000 0010h	0	2	0000 0800h
02h	Disc	0000 0011h	0	1	0000 0040h
03h	SessionMap	0000 0011h	192	1	0000 0400h
04h	C_User	0000 0012h	0	1	0000 7000h
04h	C_User	0000 0020h	0	2	0000 8000h
04h	C_User	0000 0030h	0	3	0000 8000h
00h	0	0	0	0	0
more null rows ...					
00h	0	0	0	0	0

7.2.3.2 Modify a User (Informative)

If a User is modified or erased, the Anchor table is rewritten and previous C_User rows may be referenced. The Table 22 illustrates a modification of the Table 21 Linked DVD OSPB. A user in C_User SegOrSeq 1 is modified. The modifications to the OSPB are:

- SegOrSeq of the Anchor row is incremented to reflect a new version of the Anchor table at PSARSA 40h
- Disc table and SessionMap are copied to PSARSA 41h to reflect a rewrite of the initial writable unit
- C_User SegOrSeq 1 is rewritten starting at PSARSA 42h

Table 22 — Modify a User on a DVD Linked OSPB

Anchor					
RowType	**Name**	**PSARSA**	**Offset**	**SegOrSeq**	**Size**
01h	Anchor	0000 0040h	0	3	0000 0800h
02h	Disc	0000 0041h	0	1	0000 0040h
03h	SessionMap	0000 0041h	192	1	0000 0400h
04h	C_User	0000 0042h	0	1	0000 7000h
04h	C_User	0000 0020h	0	2	0000 8000h
04h	C_User	0000 0030h	0	3	0000 8000h
00h	0	0	0	0	0
more null rows ...					
00h	0	0	0	0	0

8 Implementation Details

8.1 Admin table details

8.1.1 TPerInfo table

The TPer publishes information about itself with the TPerInfo table (Table 23). TPerInfo is defined in [SA Core] §5.4.2.1 and it is Constant in this [OSSC]. '_Maker' indicates that the device manufacturer supplies the cell content.

Table 23 — TPerInfo in OSSC TPers

UID	Bytes	GUID	Generation	Firmware Version	Protocol Version	Space For Issuance
0000 0201 0000 0001	_Maker	_Maker	_Maker	_Maker	_Maker	0

8.1.2 Properties table

The [OSSC] does not specify ComPackets, Packets, SubPackets, Transactions, nor session based timeouts. Consequently, [OSSC] has no Properties table.

8.1.3 Templates table

This table lists of all the templates in the Admin SP. The information is constant and Well Known. Thus, it is not implemented in the TPer. Table 24 shows the presumed content of this table.

Table 24 — Template table

UID	Name	Revision Number	Instances	Max Instances
0000 0204 0000 0002h	Admin	0,0,0,1	1	1
0000 0204 7777 0001h	Optical	0,0,0,1	1	1

8.1.4 SP table

The SP table (Table 25) contains information that the TPer publishes about itself. SP is defined in [SA Core] §5.4.2.6 and is Constant in this [OSSC]. '_Maker' indicates that the device manufacturer supplies cell content.

Table 25 — SP in OSSC TPers

UID	Name	ORG	Effective-Auth	DateOf-Issue	Bytes	LifeCycle-State	Frozen
0000 0205 0000 0001h	Admin	_Maker	_Maker	_Maker	_Maker	2	False

8.2 Optical tables

Clause 7, Optical Template describes the optical template table detail and their content.

8.3 UIDs used by the OSSC TPer

Any method in Table 26 may be requested from ComID 7007h. When the ComID is 7008h, the number of valid methods is smaller as indicated in the column labeled ComID 7008h.

Table 26 — UIDs of Methods implemented by OSSC TPers

Method	UID	CCS UID	ComID 7008h
Get Method	0000 0006 0000 0006h	0010h	
Set Method	0000 0006 0000 0007h	0011h	
Properties Method	0000 0000 0000 FF01h	0020h	
StartSession Method	0000 0000 0000 FF02h	0030h	√
SyncSession Method	0000 0000 0000 FF03h	0031h	√
StartTrustedSession Method	0000 0000 0000 FF04h	0032h	√
SyncTrustedSession Method	0000 0000 0000 FF05h	0033h	√
CloseSession Method	0000 0000 0000 FF06h	0034h	√
FwBegin Method	7007 0002 0BB0 0001h	0035h	
FwEnd Method	7007 0002 0BB0 0002h	0036h	
MountSV Method	7777 0001 0BB0 0001h	0040h	
SetPointer Method	7777 0006 0BB0 0000h	0050h	
EnrollBegin Method	7777 0003 0BB0 0001h	0051h	
EnrollEnd Method	7777 0003 0BB0 0002h	0052h	
ConnectBegin Method	7777 0003 0BB0 0003h	0053h	
ConnectEnd Method	7777 0003 0BB0 0004h	0054h	
EraseUser Method	7777 0006 0BB0 0005h	0055h	
Disconnect Method	7777 0006 0BB0 0006h	0056h	
Factor Method	7777 0006 0BB0 0007h	0057h	√

Table 27 — UIDs of non-methods referenced by OSSC TPers

Table/Entity	UID	CCS UID
Current SP	0000 0000 0000 0001h	0000h
Session Manager	0000 0000 0000 00FFh	0010h
TPerInfo table	0000 0201 0000 0001h	0020h
Template table	0000 0204 0000 0001h	0021h
SP table	0000 0205 0000 0001h	0022h
Drive table	7007 0002 7777 0001h	0080h
Disc table	7777 0001 7777 0001h	0081h
C_User table	7777 0003 7777 0000h	0082h

8.4 CSS Method Call/Response

All exchanges described are synchronous: a command followed by a response. IF-SEND delivers the command and IF-RECV retrieves the response. IF-RECV always returns Status in the payload header. If the response contains a response payload, the payload is appended to the response header.

8.4.1 IF-SEND and IF-RECV Payloads

The Payload for IF-SEND & IF-RECV is comprised of 3 parts, Header, Body and Footer (See Table 28 and Table 29). The Footer is always transferred, even if the TPer is not in a secure session. The MAC is always generated or evaluated within a secure session. If a MAC evaluation fails then a INVALID_SECMSG_PROPERTIES status code is reported in the Method Status during the IF-RECV.

Table 28 — CCS IF-SEND Payload

IF-SEND Payload Header Length is fixed at 8 bytes
IF-SEND Payload Body Length is dependent upon the method
IF-SEND Payload Footer Length is fixed at 32-bytes

Table 29 — CCS IF-RECV Payload

IF-RECV Payload Header Length is fixed at 12 bytes
IF-RECV Payload Body Length is dependent upon the method
IF-RECV Payload Footer Length is fixed at 32-bytes

8.4.1.1 IF-SEND Payload Parts

The IF-SEND header is shown in Table 30, the IF-SEND body is shown in Table 31, and the IF-SEND footer is shown in Table 33. All three parts are concatenated to comprise the IF-SEND payload.

Table 30 — IF-SEND header

Field Name	Payload Offset	Field Length	Field Description
SeqNumber	0	4	An incrementing counter that starts at 1 and increments with each session request. If no session has been established, this field **shall** be zeros. The TPer **shall** ignore a request with an equal or lower SeqNumber value than any previously acted-upon request. In addition, wrapping of the SeqNumber (00000000h after FFFFFFFFh) **shall** result in the session being automatically aborted.
Reserved	4	2	**Shall** be set to 0000h
AdditionalLength	6	2	This is the number of bytes remaining in the CCS Payload (Body length + Footer length). If PayLoad Length+8 is greater than the CDB Transfer Length field, the TPer **shall** ignore the request

Table 31 — IF-SEND Body

Field Name	Payload Offset	Field Length	Field Description
Body	12	BL	This content is Method specific and is detailed in the following sections. The lower limit on size is zero, the upper limit is 0xFFDC. This value is always padded to a length that is congruent to 0 mod 4.

Table 32 — IF-SEND Footer

Field Name	Payload Offset	Field Length	Field Description
MAC	12+BL	32	The SHA-256 HMAC function is used is used for secure message integrity and entity authentication. The Secret Key was passed to the TPer while starting a secure session. The MAC field is always transferred, even if a secure session does not exist.

8.4.1.2 IF-RECV Payload Parts

The IF-RECV header is shown in Table 33, the IF-RECV body is shown in Table 34, and the IF-RECV footer is shown in Table 35. All three parts are concatenated to comprise the IF-SEND payload.

Table 33 — IF-RECV header

Field Name	Payload Offset	Field Length	Field Description
SeqNumber	0	4	Shall be the SeqNumber value provided in the IF-SEND request
Method Status	4	1	Method status as specified by [SA Core]
Sense Key	5	1	If a disc access is required and an error occurs, these are the sense key, ASC, and ASCQ that would normally be reported with CHECK CONDITION status. For specific sense data lists, see [MMC-6], Error Reporting Annex.
ASC	6	1	
ASCQ	7	1	
Reserved	8	2	Shall be set to 0000h
PayloadLength	10	2	This is the number of bytes in the CCS Payload (Body length + Footer length). If CCS Payload length exceeds CDB Allocation Length field, the returned data shall be truncated to Allocation Length.

Table 34 — IF-RECV Body

Field Name	Payload Offset	Field Length	Field Description
Body	12	BL	This content is Method specific and is detailed in the following sections. The lower limit on size is zero, the upper limit is 0xFFDC. This value is always a MOD4 value.

Table 35 — IF-RECV Footer

Field Name	Payload Offset	Field Length	Field Description
MAC	12+BL	32	The SHA-256 HMAC function is used for secure message integrity and entity authentication. The Secret Key was passed to the TPer while starting a secure session. The MAC field is always transferred, even if a secure session does not exist.

8.4.2 Ignoring IF-SEND and IF-RECV

The TPer ignores an IF-SEND request by treating it as a no-operation. After all parameter data has been transferred from the Host, the command is terminated with GOOD status.

The TPer ignores an IF-RECV request by returning a response that contains only the header entirely filled with zeros. This indicates a method status of SUCCESS and no additional data beyond the header.

8.4.3 The Synchronous Nature of CCS

Once the TPer has received the entire IF-SEND payload and method execution has begun, the command **shall** be terminated with GOOD status.

As the method execution continues, the TPer constructs the response for the IF-SEND payload. If the Host requests the response prior to completion of the method execution, the TPer **shall** return only the IF-RECV payload header with Method Status set to SP_BUSY. When the Host receives an IF-RECV payload with Method Status set to SP_BUSY, the IF-RECV should be retried.

8.4.4 Session Manager Invoked Methods

8.4.4.1 SMUID.StartSession

Hosts use this method to start sessions. The Host may start a non-trusted session (Anybody Authority) or trusted session (Trusted Authorities).

Table 36 — StartSession IF-SEND body

Field	Body Offset	Field Length	Field Description
Invoking CCS UID	0	2	Session Manager CCS UID: 0010h
CCS Method UID	2	2	StartSession CCS UID: 0030h
HostSessionID	4	4	Host assigns its session ID in this method
SPID	8	8	Admin SP
Read/Write	16	2	0000h if session is read-only 0001h if session is read/write
HostExchangeAuthority	18	8	Null UID
HostExchangeCert (Length = cl)	26	cl	Host certificate that contains Host public key for the trusted session key exchange. Validated by HostExchange- Authority. See Appendix B.
Pad	26+cl	p	Minimum zero pad such that (38+cl+p) = 0 mod 4.

The StartSession response consists of only the response header.

8.4.4.2 SMUID.SyncSession

Hosts use this method to receive the TPer response to the StartSession request.

Table 37 — SyncSession IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking CCS UID	0	2	Session Manager CCS UID: 0010h
CCS Method UID	2	2	SyncSession CCS UID: 0031h

Table 38 — SyncSession IF-RECV Body

Field	Body Offset	Field Length	Field Description
HostSessionID	0	4	Return session ID supplied in StartSession
SPSessionID	4	4	TPer side session ID, assigned by the TPer.
SPExchangeCert (Length = cl)	8	cl	SP certificate that contains TPer public key for the trusted session key exchange. Validated by HostExchangeAuthority. See Appendix B.
Pad	8+cl	p	Minimum zero pad such that (38+cl+p) = 0 mod 4.

8.4.4.3 SMUID.StartTrustedSession

Complete opening a trusted session for secure messaging.

Table 39 — StartTrustedSession IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking CCS UID	0	2	Session Manager CCS UID: 0010h
CCS Method UID	2	2	StartTrustedSession CCS UID: 0032h
HostSessionID	4	4	Host session ID assigned in StartSession
SPSessionID	8	4	SP session ID assigned in SyncSession
HostSessionKey	12	256	Key and IV used by SP to decrypt subsequent messages from Host. These fields are concatenated and then encrypted with the TPer Public Key.
HostSessionIV			

The StartTrustedSession response consists of only the response header.

8.4.4.4 SMUID.SyncTrustedSession

Hosts use this method to receive the TPer response to the StartTrustedSession request.

Table 40 — SyncTrustedSession IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking CCS UID	0	2	Session Manager CCS UID: 0010h
CCS Method UID	8	2	SyncTrustedSession CCS UID: 0033h

Table 41 — SyncTrustedSession IF-RECV Body

Field	Body Offset	Field Length	Field Description
HostSessionID	0	4	Host session ID assigned in StartSession
SPSessionID	4	4	SP session ID assigned in SyncSession
SPSessionKey	8	256	Key and IV used by Host to decrypt subsequent messages from TPer. These fields are concatenated and then encrypted with the Host Public Key.
SPSessionIV			

8.4.4.5 SMUID.CloseSession

This method closes the currently open session. If a user is connected, the user **shall** be disconnected.

Table 42 — CloseSession IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking CCS UID	0	2	Session Manager CCS UID: 0010h
CCS Method UID	2	2	SyncTrustedSession CCS UID: 0034h
HostSessionID	4	4	Host session ID assigned in StartSession

The CloseSession response consists of only the response header.

8.4.4.6 SMUID.FwBegin

The FwBegin declares the start of a secure firmware download to the TPer. All Optical Disc Management method sequences and User Management method sequences must be completed before entering this phase. All writes and reads to the media by the host **shall** be prevented until the firmware download is complete. Any connections **shall** be disconnected.

The actual firmware download process is TBD.

Table 43 — FwBegin IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking CCS UID	0	2	Session Manager CCS UID: 0010h
CCS Method UID	2	2	FwBegin CCS UID: 0035h

The FwBegin response consists of only the response header and footer.

8.4.4.7 SMUID.FwEnd

The FwEnd method declares the end of the code download process. If the firmware is successfully authenticated status **shall** be returned with that indication. The firmware **shall** then be saved and the TPer **shall** execute a hard reset.

Table 44 — FwEnd IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking CCS UID	0	2	Session Manager CCS UID: 0010h
CCS Method UID	2	2	FwEnd CCS UID: 0036h

The FwEnd response consists of only the response header and footer.

8.4.5 C_User Invoked Methods

The C_User table rows are not directly accessible by the host. The TPer has a C_User row pointer that the Host sets in order to "select" a C_User row for operations.

C_User Row 0 always contains the Initializer. The Initializer is the first enrolled user. The Initializer is the only user authorized to enroll new users or delete existing users. A Host **shall** be connected as the Initializer to perform these functions.

8.4.5.1 C_User.SetPointer

The Host requests that the TPer set its C_User row pointer to a specific C_User row. The Row content is always returned. Status of the operation is returned in the IF-RECV header.

Table 45 — SetPointer IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking UID	0	2	C_user CCS UID: 0082h
Method UID	2	2	SetPointer CCS UID: 0050h
Operation	4	4	0 = Read the current C_User entry. 1 = Point to next C_User row then read entry. 2 = Point to the C_User Initializer row (Row 0) 3 = Point to C_User row associated with supplied Row index
Addressed row	8	2	Row index, only used for operation 3.
Reserved	10	2	Reserved

Table 46 — SetPointer IF-RECV Body

Field	Body Offset	Field Length	Field Description
RowX	0	2	C_User row index
UserName	8	64	Byte array
Is_Valid	72	1	0 = Row is not initialized (following fields are not valid). 1 = Row is populated (following fields are valid).
Authority	73	1	0 = Initializer (Row 0) 1 = Common user
Reserved	74	2	Byte array

8.4.5.2 C_User.EnrollBegin

The Initializer user is the first user enrolled. The Initializer is the only user that is permitted to add a new common user. Enroll C_user is used to insert a new C_User row in C_Users. The TPer selects the C_User location.

The LocalHost should verify that every UserName is unique.

Table 47 — EnrollBegin IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking UID	0	2	C_User CCS UID: 0082h
Method UID	2	2	EnrollBegin ID Abbreviation: 0051h
Reserved	4	3	
NumFactors	7	1	Number of authentication factors.
UserName	8	64	Byte array

The EnrollBegin response consists of only the response header.

8.4.5.3 C_User.EnrollEnd

The Host uses this method to declare the end of the Enrollment process. The TPer **shall** set status in the EnrollEnd response header that indicates either success or failure.

Table 48 — EnrollEnd IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking CCS UID	0	2	C_User CCS UID: 0082h
CCS Method UID	2	2	EnrollEnd CCS UID: 0052h

The EnrollEnd response consists of only the response header.

8.4.5.4 C_User.Erase

The Initializer is the only user that is permitted to erase a C_User row. The Host should first request SetPointer to point to the C_User row to erase. The Erase method **shall** erase the pointed C_User The Initializer cannot be erased.

Table 49 — Erase IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking CCS UID	0	2	C_User CCS UID: 0082h
CCS Method UID	2	2	Erase CCS UID: 0053h

The Erase response consists of only the response header.

8.4.5.5 C_User.ConnectBegin

If any user is connected, the invocation of ConnectBegin includes an implied Disconnect of the currently connected user. The Host should first set the C_User row pointer prior to requesting ConnectBegin. The process is associated with pointed the C_User row. A successful connection is dependent upon all authentication factors being supplied.

Table 50 — ConnectBegin IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking CCS UID	0	2	C_User CCS UID: 0082h
CCS Method UID	2	2	ConnectBegin CCS UID: 0054h

The ConnectBegin response consists of only the response header.

8.4.5.6 C_User.ConnectEnd

The Host uses this method to declare the end of the connection process. The TPer **shall** set status in the ConnectEnd response header that indicates either success or failure.

Table 51 — ConnectEnd IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking CCS UID	0	2	C_User CCS UID: 0082h
CCS Method UID	2	2	ConnectEnd CCS UID: 0055h

The ConnectEnd response consists of only the response header.

8.4.5.7 C_User.Factor

The factor is applied towards the current C_user row. These factors may be supplied within the existing local host secure session or ExternalHost secure session.

Table 52 — Factor IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking CCS UID	0	2	C_User CCS UID: 0082h
CCS Method UID	2	2	Factor CCS UID: 0057h
Pass-Code	4	32	Authentication Factor

The Factor response consists of only the response header and footer.Disc Invoked Methods

8.4.5.8 Disc.Get

This command retrieves information from the Disc Table.

Table 53 — Disc.Get IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking CCS UID	0	2	Disc table CCS UID: 0081h
CCS Method UID	2	2	Get CCS UID

Table 54 — Disc.Get IF-RECV Body

Field	Body Offset	Field Length	Field Description
CryptoType	0	1	CryptoType field of Disc table
HSP_valid	1	1	Host Supplied Parameters valid (see Table 15)
HSP	2	64	Host Supplied Parameters (see Table 15)
DiscUniqueID	66	16	Disc Unique Identifier (see Table 15)
HostMAC	82	32	HostMAC field of Disc table

8.4.5.9 Disc.Set

This command updates information in the Disc Table. Only writable fields are updated. Contents of non-writable fields are ignored. See Table 15.

Table 55 — Disc.Get IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking UID	0	2	Disc table CCS UID: 0081h
Method UID	2	2	Set CCS UID: 0011h
CryptoType	4	1	If current CryptoType cell = 0, and proposed value is valid and non-zero, write proposed value into cell. If current CryptoType cell ≠ 0, cell is not written.
HSP_valid	5	1	Host Supplied Parameters valid (see Table 15)
HSP	6	64	Host Supplied Parameters (see Table 15)
HostMAC	70	32	If current HostMAC value is 32 zeros, write proposed value; otherwise cell is not written.

The Disc.Get response consists of only the response header and footer.

8.4.5.10 Disc.MountSV

If SV is not mounted, dismount current volume and MountSV. For MMC devices this means:

1. Post a disc removal event to be detected by the GET EVENT STATUS NOTIFICATION command,
2. Post a new media event also to be detected by the GET EVENT STATUS NOTIFICATION command,

For Hosts that poll with the GET EVENT STATUS NOTIFICATION command in order to learn of asynchronous drive events, this represents a media change.

Table 56 — MountSV IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking CCS UID	0	2	Disc CCS UID: 0081h
CCS Method UID	2	2	MountSV CCS UID: : 0040h

The MountSV response consists of only the response header.

8.4.6 Drive Invoked Methods

8.4.6.1 Drive.Get

This command retrieves information from the Disc Table.

Table 57 — Drive.Get IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking CCS UID	0	2	Drive table CCS UID: 0080h
CCS Method UID	2	2	Get CCS UID: 0010h

Table 58 — Drive.Get IF-RECV Body

Field	Body Offset	Field Length	Field Description
DriveAES	0	1	Content from DriveAES cell
TrustedMode	1	1	Content from TrustedMode cell
Pad	2	2	Pad to 0 MOD(4) boundary.

8.4.6.2 Drive.Set

This command updates information in the Disc Table. Only writable fields are updated. Contents of non-writable fields are ignored. See Table 15.

Table 59 — Drive.Get IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking CCS UID	0	2	Drive CCS UID: 0080h
CCS Method UID	2	2	Set CCS UID: 0011h
DriveAES	4	1	Dummy data that is not written
TrustedMode	5	1	If current TrustedMode cell = 0, and proposed value is not zero, write 1 into cell. If current TrustedMode cell ≠ 0, cell is not written.
pad	6	2	Pad to 0 MOD(4) boundary.

The Disc.Get response consists of only the response header and footer.

8.4.7 Admin Invoked Methods

8.4.7.1 SP.Get

This command retrieves information that the TPer publishes about itself. The purpose of the Admin Template is to provide to the Admin SP the capability to optionally Issue additional SPs and to maintain information about the TPer. The optical SSC does not permit SP issuance, so most values are null.

Table 60 — Get IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking CCS UID	0	2	SP table CCS UID
CCS Method UID	2	2	Get CCS UID: 0010h

Table 61 — SP.Get IF-RECV Body

Field	Body Offset	Field Length	Field Description
Name	0	32	Left justified Unicode text, padded with zeros with the content set to “Admin”.
ORG	32	8	_maker defined
EffectiveAuth	40	32	_maker defined
DateOfIssue	72	4	The date of Issuance [Year(2 bytes), Month(1 byte), Day(1 byte)]
Bytes	76	8	Size of the SP; _maker defined
LifeCycleState	84	1	2 = issued-frozen
Frozen	85	1	0 = False

8.4.7.2 TPerInfo.Get

This command retrieves information that the TPer publishes about itself.

Table 62 — TPerInfo.Get IF-SEND Body

Field	Body Offset	Field Length	Field Description
Invoking CCS UID	0	2	TPerInfro table CCS UID
CCS Method UID	2	2	Get CCS UID: 0010h

Table 63 — TPerInfo.Get IF-RECV Body

Field	Body Offset	Field Length	Field Description
Bytes	0	8	The size in bytes of the TPer's entire protected storage area. _maker defined
GUID	8	8	TPer's globally unique serial number (See [SA Core]); _maker defined
Generation	16	4	Generation number of the volume; _maker defined
Firmware Version	20	4	Manufacturer-defined revision number of the TPer firmware; _maker defined
Protocol Version	24	4	Revision number of the interface messaging protocol; _maker defined
SpaceFor-Issuance	28	8	0 =Amount of available bytes remaining for issuance.

9 User Scenarios (Informative)

9.1 Role of LocalHost

The process of authenticating and authorizing a user is defined as the connect process in this [OSSC].

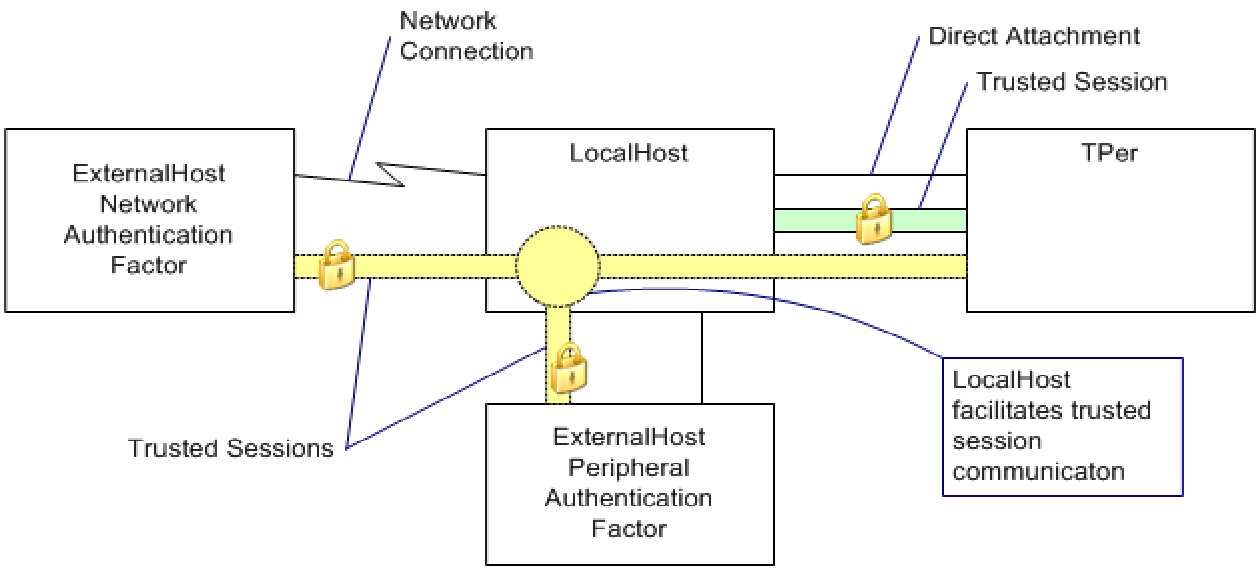
Authentication may require one or more factors. Each factor is known by exactly one Host, either LocalHost or some ExternalHost.

LocalHost provides the initial authentication factor. If other authentication factors are required, they may be provided from ExternalHost(s). Each ExternalHost **shall** provide its factor in a trusted session with the TPer. LocalHost facilitates communication with each ExternalHost, but message semantics are known only to trusted session endpoints. The LocalHost authentication factor is required and is always provided to the Tper first. Each user may have a different set of factors.

Examples of what a factor may be are:

- Something the user knows (Pass-code)
- Something the user has (security token, personal computer)
- A physical characteristic of the user (biometrics)
- A colleague of the user (network password recovery, dual login)
- Location of the user (network address filter)
- Time of the user's connection attempt (network time service)
-

Figure 6 — Authentication Logical Communication Paths



9.2 Procedure to Enroll the Initializer

The Initializer is the disc's crypto-officer and is identified only by being the first user enrolled. A write-once disc **shall** be empty: either blank or minimally formatted with a blank user data area. A rewritable disc **shall** be empty: either blank or minimally formatted.

Suppose:

- UserX is a LocalHost user that wishes to Enroll,
- UserX has determined an acceptable UserName,
- N is the number of factors that are to be required for authentication of UserX,
- H_J (J = 1,..,N) are Hosts, where H_1 is LocalHost and H_2,..H_N are ExternalHosts.. For each J, UserX has arranged for exactly one authentication factor, F_J, from each Host, H_J.

The following procedure is required:

1. LocalHost establishes a trusted session with the TPer with StartSession[ ] and StartTrustedSession[ ] at ComID 7007h.
2. LocalHost should invoke disc.set to select encryption type.
3. LocalHost invokes SetPointer to point to the Initializer row. If the Initializer row has Valid status, then the disc has been initialized and a new Initializer is not permitted. If new Initializer is not permitted, the enrollment process **shall** be terminated and all sessions **shall** be closed.
4. LocalHost invokes EnrollBegin, providing a UserName. If the User table row pointer is not at the first row, the enrollment process **shall** be terminated and all sessions **shall** be closed.

   Otherwise, the TPer executes EnrollBegin[ ] by:

   - TPer points to a C_User row that has the not valid (empty) status.
   - UserName is copied to the pointed user row.
   - The DK_accumulator is initialized to a Hash of (RowX || UserName).
   - Factor_count is set to 0.
5. *LocalHost performs steps a-d for each authentication factor F_J, J = 1,2,..,N*.
   a. If H_J is an ExternalHost, then LocalHost requests ExternalHost, H_J to authenticate on ComID 7008h. H_J establishes a trusted session with the TPer through LocalHost.
   b. H_J invokes Factor, providing its authentication factor, F_J. The TPer executes Factor[F_J] by applying DeriveKey(F_J) and XOR'ing the result in DK_accumulator.
   c. Factor_count is incremented.
   d. If H_J is an ExternalHost, then H_J returns the result of Factor[ ] to LocalHost.
6. LocalHost invokes EnrollEnd. The TPer executes EnrollEnd[ ] as follows:
   a. Factor_count is stored into the Nfactors cell of the User row.
   b. MakeEAC() is applied, setting the EAC cell.
   c. A PSAkey is generated and stored in the user row. The FDEkey is generated and stored in the Disc table. The CRPs in the Disc table are encrypted with the PSAkey. The EAC is calculated and stored in the user row. The secure parts of the user row are encrypted using DK_accumulator as the AES-128 key. The user row state variable (IsValid) is set to Valid. The result from EnrollEnd[ ] indicates the result of this procedure.

If at any point during the enrollment process, the TPer reports any method status other than SUCCESS, the TPer **shall** terminate the enrollment process and close all open sessions.

9.3 Procedure to Enroll a User

Only the Initializer may Enroll a new user. Enrolling a new user with N authentication factors is similar to connection authentication with N factors. Suppose:

- UserX is a LocalHost user that wishes to Enroll,
- UserX has determined an acceptable UserName,
- N is the number of factors that are to be required for authentication of UserX,
- H_J (J = 1,..,N) are Hosts, where H_1 is LocalHost and H_2,..H_N are ExternalHosts.. For each J, UserX has arranged for exactly one authentication factor, F_J, from each Host, H_J.

The following procedure is required:

1. The Initializer is connected according to 9.4. This makes PSAkey available for enrollment.
2. LocalHost invokes EnrollBegin, providing a UserName. If there is no connection or the current connection is not the Initializer, the connection process **shall** be terminated and the session **shall** be closed. Otherwise, the TPer executes EnrollBegin[ ] by:
   - TPer points to a C_User row that has the not valid (empty) status.
   - UserName is copied to the pointed user row.
   - The DK_accumulator is initialized to a Hash of (RowX || UserName).
   - Factor_count is set to 0.
3. *LocalHost performs steps a-d for each authentication factor F_J, J = 1,2,..,N*.
   a. If H_J is an ExternalHost, then LocalHost requests ExternalHost, H_J to authenticate on ComID 7008h. H_J establishes a trusted session with the TPer through LocalHost.
   b. H_J invokes Factor, providing its authentication factor, F_J. The TPer executes Factor[F_J] by applying DeriveKey(F_J) and XOR'ing the result in DK_accumulator.
   c. Factor_count is incremented.
   d. If H_J is an ExternalHost, then H_J returns the result of Factor[ ] to LocalHost.
4. LocalHost invokes EnrollEnd. The TPer executes EnrollEnd[ ] as follows:
   a. Factor_count is stored into the Nfactors cell of the User row.
   b. MakeEAC() is applied, setting the EAC cell.
   c. The secure parts of the row are encrypted using DK_accumulator as the AES-128 key. The user row state variable (IsValid) is set to Valid. The result from EnrollEnd[ ] indicates the result of this procedure. Recording the appropriate OSPB may be deferred.

If at any point during the enrollment process, the TPer reports any method status other than SUCCESS, the TPer **shall** terminate the enrollment process and close all open sessions.

9.4 Procedure to Connect a User

Suppose:

- UserX is a LocalHost user who wishes to connect to the disc's Secure Volume,
- N is the number of factors that are required for authentication of UserX,
- H_J (J = 1,..,N) are Hosts, where H_1 is LocalHost and H_2,..H_N are also Hosts, some of which may be ExternalHosts.

For UserX and for each J, H_J possesses exactly one authentication factor, F_J.

The following procedure is required for connection:

1. LocalHost establishes a trusted session with the TPer with StartSession[ ] and StartTrustedSession[ ] at ComID 7007h.
2. LocalHost invokes SetPointer iteratively to select the C_User row that contains the UserName of UserX.
3. LocalHost invokes ConnectBegin. The TPer executes ConnectBegin[ ]: DK_accumulator is initialized to a Hash of (RowX || UserName). Factor_count is set to 0.
4. *LocalHost performs steps a-c for each authentication factor F_J, J = 1,2,..,N*.
    a. If H_J is an ExternalHost, then LocalHost requests ExternalHost, H_J to authenticate on ComID 7008h. H_J establishes a trusted session with the TPer through LocalHost.
    b. H_J invokes Factor, providing its authentication factor, F_J. The TPer executes Factor[F_J] by applying DeriveKey(F_J) and XOR'ing the result in DK_accumulator. Factor_count is incremented.
    c. If H_J is an ExternalHost, then H_J returns the result of Factor[ ] to LocalHost.
5. LocalHost invokes ConnectEnd. The TPer executes ConnectEnd[ ] by using DK_accumulator to decrypt the secure parts of the pointed C_User row. CheckEAC( ) is then applied to the decrypted C_User row. If CheckEAC( ) fails or if Factor_count ≠ Nfactors, then the user has not been validated and the method status of ConnectEnd[ ] is ACCESS_DENIED. Otherwise, the result of ConnectEnd[ ] is SUCCESS and all cells in the decrypted C_User row are available. The TPer uses the PSAkey to decrypt the Disc table. That result contains the FDEkey and DiscIVbase.

If at any point during the connection process, the TPer reports any method status other than SUCCESS, the TPer **shall** terminate the connection process and close all open sessions.

9.5 Procedure to Erase a User

Erasing a user is only available to the Initializer.

1. The Initializer is connected according to 9.4.
2. LocalHost uses SetPointer to select the C_User row to erase.
3. LocalHost invokes Erase. If the Initializer has not set up the Trusted Session and connection, the connection and session **shall** be aborted. The TPer executes the Erase by marking the IsValid entry in the pointed C_User row to not valid. The TPer may take action to ensure that the set of Valid C_User rows and the set of Empty C_User rows form two contiguous areas of the C_User table. Re-recording the appropriate OSPB may be deferred.

10 Security Evaluation (Informative)

10.1 Security Protocols

This [OSSC] requires two security protocols:

- a public key exchange protocol to establish trust between Trusted Authorities and the TPer and to establish session keys for secure messaging
- a key derivation, multi-factor authentication protocol to Connect users to the Secure Volume

The public key protocol uses Session Manager methods as defined in [SA Core] and this [OSSC]. User Connection is defined in this [OSSC]. User Connection occurs within secure messaging that was established with the public key exchange protocol.

10.2 Sensitive Security Parameters

[FIPS140-3] defines **C**ritical **S**ecurity **P**arameters (CSP's) as secret and private cryptographic keys and authentication data (Passcodes) whose disclosure or modification can compromise the security of the cryptographic module. It defines public security parameters as public information whose modification can compromise the security of the cryptographic module. The union of these two sets is defined as **S**ensitive **S**ecurity **P**arameters (SSP's).

[NIST SP800-57] provides guidance on the security strength of cryptographic algorithms, key sizes, and their expected lifetimes. The SSPs used in this [OSSC] are given in the following table.

Table 64 — Security Strength and Lifetime

Cryptographic Function	Security Strength	Usage	Security Lifetime
AES-128 CBC	128 bits	bulk data encryption message confidentiality table confidentiality	beyond 2030
SHA-256 hash-only	128 bits	key derivation function random number generation	beyond 2030
SHA-256 HMAC	128 bits	message integrity entity authentication check	beyond 2030
RSA-2048	112 bits	Trusted Authority authenticaton	through 2030

10.2.1 Asymmetric Security Parameters

Asymmetric security parameters are used with authentication of Trusted Authorities. All optical TPer's store a root public key in a manner that resists malicious modification.

10.2.2 Symmetric Security Parameters

CSP's that are used for user Connection: DK_accumulator, EAC, PSAkey, and FDEkey. All are transitory until FDEkey is written to a bulk encryption register; this register should be implemented as write only. DK_accumulator likely has the longest life of the CSP's. However, it is protected by the one-way SHA-256 algorithm used in DeriveKey( ) and C_User.Nfactors. Implementers should erase all CSP's after they have served their purpose.

10.3 Threat Model

Four mechanisms provide the security foundation:

- Mutual authentication: chains trust among participating entities; see [SA Core] §3.4.4.5, Starting Sessions
- Secure messaging: provides confidentiality, integrity and sequencing; see [SA Core] §3.2.3.5.1, Secure Messaging Packet Format
- User authentication: provides a mechanism such that organizational security policy has the final responsibility for security; this [OSSC] provides cryptographic protection of CSP's on the disc and does not store CSP's on the device
- Multi-factor authentication: when deployed as a security policy, reduces the risk of single point failure

Mutual authentication, secure messaging and user authentication are enforced by the TPer; organizational security policy is responsible for configuring multi-factor authentication.

10.3.1 Missing disc attack

An adversary gains possession of a disc. This [OSSC] claims to mitigate this attack with FDE and user authentication.

10.3.2 Data breach attack

An adversary gains access to sensitive data that is supposed to be kept secret at participating entities (LocalHost, ExternalHosts, and local security tokens). This [OSSC] claims to mitigate this attack with multi-factor authentication.

10.3.3 Eavesdropping attack

An adversary snoops communications to learn useful information. This [OSSC] claims to mitigate this attack with secure messaging.

10.3.4 Replay attack

An adversary records messages that were sent in past communications and re-sends them at a later time. This [OSSC] claims to mitigate this attack with mutual authentication and secure messaging.

10.3.5 Man-in-the-middle attack

An adversary intercepts the messages sent between entities and replaces them with its own messages. This [OSSC] claims to mitigate this attack with mutual authentication and secure messaging.

10.3.6 Password-guessing attack

An adversary repeatedly picks a password from a dictionary and tries to use it in order to impersonate the user. This [OSSC] claims to mitigate this attack with multi-factor authentication.

10.3.7 VolumeZero attacks

VolumeZero is the clear-text volume that Hosts may mount prior to User authorization.

10.3.7.1 Trojan attack

An adversary injects malicious software in the clear-text VolumeZero. This [OSSC] provides a mechanism, not a complete solution, to mitigate this attack. Disc{ } includes the HostMac cell that LocalHost may use to authenticate VolumeZero. Security policy (disable auto-mount) and LocalHost software have the ultimate responsibility for mitigating this attack.

10.3.7.2 Smuggler attack

An adversary desires to smuggle sensitive information out of an organization. A comprehensive organization security policy with multiple mechanism layers is required to mitigate this attack. The lowest mechanism provided by this [OSSC] is to enable TrustedMode (=1) to remove this bypass capability. This restricts disc creation to only TCG discs. It also puts VolumeZero under TPer access

control. An organization has several options to build trust into the LocalHost application, including the multi-factor authentication mechanism. Other measures to mitigate this attack are beyond the scope of this [OSSC].

10.3.8 Attacks not defended

Attacks that are not documented in this [OSSC] should be considered undefended. Attacks that are undefended include but are not limited to:

1) This [OSSC] does not provide data security after user authorization. Once a user is authorized, all user data is accessible.
2) This [OSSC] does not defend denial of service attacks; this is the responsibility of the LocalHost.
3) This [OSSC] does not defend against cipher-text modification attacks. Binding sectors or collections of co-related sectors for the purpose of detecting data tampering is the responsibility of the recording application.

Appendix A. [SA Core] Deviations

This [OSSC] has deviations from the [SA Core].

Topic	Reference	Comments
Stream Encoding	§3.2.3	CCS is the implementation for Protocol ID 6. See Level 0 Discovery.
ComID	§3.3.2-3.3.3	Static ComID values are used. ComID 7007h and 7008h are used by default and are considered in the "Issued" state after POR GET_COMID, HANDLE_COMID_REQUEST, GET_COMID_RESPONSE are not supported
SP Issuance, et al	§3.4.3	No host issuance is supported
Sessions, Methods, Transactions	§3.4.4	A maximum of one open session is supported per ComID. In CCS a method execution request and its response constitute a transaction. Transactions are not nested.
Life Cycle of SPs	§4	Only support Issued (Enabled). All other Life Cycle states are not supported

Appendix B. Disc Interchange and Compatibility (Informative)

Optical discs are removable media. The optical drive/disc and PC industry has put a lot of effort into ensuring interchangeability and compatibility among all suppliers of discs, drives, and PC's. In order to provide a similar level of compatibility and interchangeability between drive and PC suppliers who implement data security according to this SSC it is important that several of the security parameters are standardized across all suppliers.

The items that need to be agreed upon are:

1) Constants for DeriveKey(), MakeEAC(), and CheckEAC()

   Standardizing these constants enables interchangeability across different drive manufacturers. Additionally, some organizations may arrange for unique values for these constants, thus ensuring discs written inside their organizations are not readable by drives outside their organizations.

2) Seeding for random number generation

   This Optical SSC specifies three levels of keyed protection, where two levels are generated randomly. Unique seeding avoids identical sequences from pseudo-random number generators.

3) TPer Certificates

   The certificate contains a unique identifier for the Optical Disc Drive group and consequently, the standard/unique parameter set.

4) Host Certificates

   A common source for these certificates permits wide use for standard parameter sets while maintaining the ability to restrict access to closed groups.

Section 5.2.3, Key Derivation and Entity Authentication, describes the Salt and HMAC used for key derivation and validation. There are two defined sources for these parameters:

1. A standard set of parameters is used by all Optical Disc Drive manufacturers. These parameters are secured within the drives and are never sent across the interface to the host computer. This enables interchange among all Optical Disc Drives that implement this SSC.
2. The LocalHost is permitted to supply these parameters and they are stored in the Disc Table (see 7.1.2). Given the large installed base of Optical Disc Drives that precede this SSC, this method provides the possibility to create a software reader. This method is considered less secure since the Salt and HMAC keys are stored in the clear on the optical disc and the user data will be decrypted by host software instead of by secure hardware in the optical drive. However, for some organizations the tradeoff in security will be acceptable and allow them to leverage the large installed base of non-TCG drives for readers.